

# APLIKASI SISTEM PERANGKAT LUNAK MENGGUNAKAN ALGORITMA ANT UNTUK MENGATUR PENJADWALAN KULIAH

*(The Software Application System Using Ant Algorithm for Scheduling of Courses)*

Hoga Saragih<sup>1</sup>, Genrawan Hoendarto<sup>2</sup>, Bobby Reza<sup>3</sup>, Didik Setiyadi<sup>4</sup>

<sup>1</sup>Fakultas Teknik dan Ilmu Komputer – Universitas Bakrie

<sup>2</sup>Program Studi Sistem Informasi – STMIK Widya Dharma

<sup>3,4</sup>Program Pasca Sarjana – STMIK Eresha

<sup>1</sup>hogasaragih@gmail.com, <sup>2</sup>genrawan@yahoo.com, <sup>3</sup>bobby.reza@eresha.ac.id,

<sup>4</sup>didik.setiyadi@eresha.ac.id

## Abstrak

Algoritma *Ant* merupakan salah satu dari algoritma *swarm intelligence* yang dapat digunakan untuk menyelesaikan masalah NP-Hard. Penjadwalan mata kuliah dan dosen yang dilakukan secara rutin setiap semester dapat diselesaikan dengan metode eksak dan metode metaheuristik. Penjadwalan melibatkan beberapa batasan yang berbeda antar institusi, dalam tulisan ini mengambil studi kasus di STMIK Widya Dharma. Jenis algoritma *Ant* yang digunakan adalah *Rank Base Ant System* yang sesuai untuk penjadwalan berdasarkan peringkat, yaitu tingkat prioritas dosen, sedangkan parameter lainnya adalah mata kuliah, ruangan, dan waktu. Diharapkan dengan adanya perangkat lunak yang dianalisis dengan algoritma *Ant*, dapat memberikan solusi masalah penjadwalan.

**Kata Kunci:** penjadwalan perkuliahan, algoritma *Ant*,  $AS_{Rank}$

## Abstract

*Ant algorithm is one of Swarm Intelligence algorithms that can be used to solve NP-Hard problems. The scheduling of courses and lecturers which are done routinely each semester can be solved by exact methods and heuristic methods. Each institution has its own constraints in arranging the schedule. This paper discusses a case study in STMIK Widya Dharma. The type of ant algorithm used is the Rank Base Ant System suitable for scheduling based on ranking, which is the lecturers' priority level. The other parameters are subjects, space, and time. It is expected that the software being analyzed with Ant algorithms can provide solutions to the scheduling problem.*

**Keywords:** university timetable course, An algorithm,  $AS_{Rank}$

## 1. PENDAHULUAN

Sebuah jadwal merupakan sekumpulan dari pertemuan pada waktu tertentu. Sebuah pertemuan adalah kombinasi dari sumber daya (ruangan, orang, dan lainnya), dimana beberapa diantaranya ditentukan oleh masalah dan beberapa mungkin dialokasikan sebagai bagian dari pemecahan [1]. Masalah penjadwalan kuliah merupakan masalah NP-hard (*nondeterministik polynomial – hard*), yang artinya waktu yang diperlukan untuk perhitungan pencarian solusi meningkat secara eksponensial dengan bertambahnya ukuran masalah [2]. Sistem penjadwalan otomatis telah digunakan secara luas dalam institusi pendidikan untuk menghasilkan jadwal yang efisien dalam

penyusunan [2]. Prosedur penjadwalan otomatis dapat dikelompokkan dalam empat kategori, yaitu:

- Metode sekuensial, seperti pewarnaan *graph*,
- Metode *clustering*,
- Metode berbasis batasan, seperti *programming integer*,
- Metode perhitungan intelijen, seperti algoritma genetik, *simulated annealing*, *tabu search*, *neural network*, *artificial immune system*, dan algoritma *Ant*.

Algoritma *Ant* merupakan salah satu dari teknik yang paling sukses dalam hal penjadwalan menurut [3] dan [4], terutama diaplikasikan dalam TSP (*travelling salesman problem*).

Generasi pertama program masalah penjadwalan dengan komputer dikembangkan pada awal tahun 1960 yang berusaha mengurangi pekerjaan administratif [5], [6]. Peneliti telah mengusulkan berbagai pendekatan penjadwalan dengan menggunakan metode berdasarkan batasan-batasan, pendekatan berdasarkan populasi, seperti algoritma genetik, algoritma *Ant*, algoritma Memetic, metode meta heuristik seperti *tabu search*, *simulated annealing* dan *great deluge*, *variable neighbourhood search* (VNS), *hybrid meta-heuristics* dan *hyper heuristic approaches*, dan lain sebagainya [7].

## 2. KONSEP PENJADWALAN

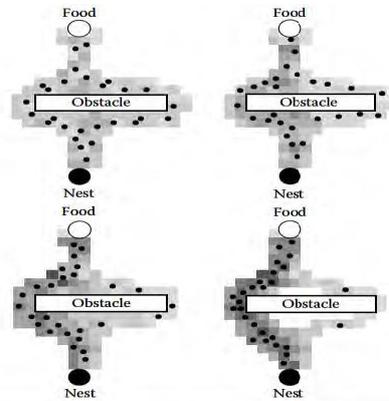
Macam-macam batasan (*constraint*) dalam masalah penjadwalan perkuliahan menurut [8] seperti berikut ini:

- *Edge constraint*, dua kejadian tidak boleh menempati *slot* waktu yang sama.
- *Ordering constraint*, batasan yang mengatur urutan kejadian.
- *Event spread constraint*, batasan yang mengatur penyebaran kejadian pada suatu penjadwalan.
- *Preset specification and exclusion*, batasan yang menentukan terlebih dahulu *slot* waktu yang akan digunakan oleh suatu kejadian sebelum proses pencarian solusi dilakukan.
- *Capacity constraint*, batasan yang berhubungan dengan kapasitas suatu ruangan.
- *Hard and soft constraint*. *Hard constraint* adalah batasan yang tidak boleh dilanggar sama sekali dan *soft constraint* yang diusahakan semaksimal mungkin untuk dipenuhi.

Dengan adanya batasan-batasan tersebut, maka masalah penjadwalan lebih cocok diselesaikan dengan metode heuristik dibandingkan metode eksak. Algoritma *Ant* merupakan pendekatan metaheuristik berdasarkan populasi yang telah sukses diaplikasikan pada banyak masalah optimisasi, khususnya *constraint satisfaction problems/CSPs* (masalah batasan kepuasan). Idenya menggunakan semut buatan untuk melacak daerah yang menjanjikan dari ruang pencarian dengan meletakkan jejak feromon. Informasi feromon ini digunakan untuk memandu pencarian sebagai heuristik dalam memilih nilai-nilai untuk ditujukan ke variabel. Pertama, digambarkan algoritma *Ant* dasar untuk pemecahan CSP dan ditunjukkan bagaimana dapat ditingkatkan melalui penggabungan dengan teknik pencarian lokal (*local search*). Pemecahan masalah batasan kepuasan melibatkan pencarian nilai yang sesuai untuk variabel yang memenuhi seperangkat batasan [9].

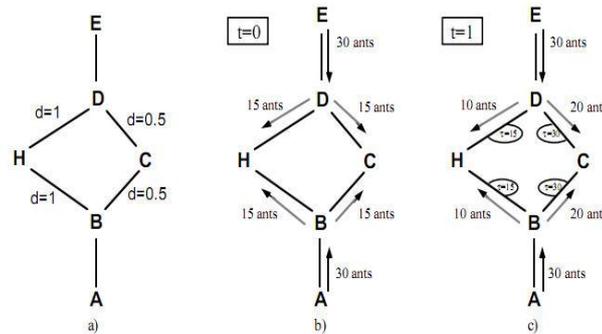
Algoritma semut adalah *bioinspired metaheuristic*, mempunyai sekelompok khusus yang berusaha menyamai karakteristik kelakuan dari serangga sosial, yaitu koloni semut. Kelakuan dari tiap pelaku dalam meniru kelakuan dari semut hidup dan bagaimana mereka berinteraksi satu dengan lainnya agar dapat menemukan sumber makanan dan membawanya ke koloni mereka dengan efisien. Selama berjalan tiap semut

mengeluarkan feromon, dimana semut lainnya sensitif dengan feromon tersebut sehingga memberikan harapan untuk mengikuti jejaknya. Lebih atau kurang intensitasnya tergantung pada konsentrasi dari feromon. Setelah beberapa waktu, jalur terpendek akan lebih sering diikuti dan feromonnya menjadi jenuh.



Gambar 1. Perubahan konsentrasi feromon

Untuk penjelasan lebih lanjut perhatikan Gambar 2, yang merupakan kemungkinan sebagai interpretasi dari Gambar 1. Untuk memperjelas dimisalkan jarak antara D dan H, B dan H, B dan D dengan melalui C adalah sama dengan satu, dan anggaphlah C berada setengah jalan antara D dan B. Perhatikan apa yang terjadi pada interval waktu reguler terdiskritisasi:  $t = 0, 1, 2, \dots, n$ . Misalkan 30 semut baru datang dari A ke B, dan 30 semut dari E ke D pada setiap satuan waktu. Misalkan setiap semut masing-masing berjalan pada kecepatan 1 per satuan waktu, dan sambil berjalan semut meletakkan jejak feromon dengan intensitas 1 pada waktu  $t$ . Untuk menyederhanakan contoh, anggaphlah feromon akan menguap sepenuhnya dan seketika di tengah interval waktu berturut-turut ( $t + 1, t + 2$ ). Pada waktu  $t = 0$  belum ada jejak, dan misalkan 30 semut ada di B dan 30 semut ada di D. Pilihan semut-semut untuk melalui jalan mana yang harus dilewati adalah acak. Oleh karena itu, anggaphlah masing-masing 15 semut dari setiap *node* akan pergi ke arah H dan 15 lainnya ke C. Pada saat  $t = 1$ , maka baru 30 semut yang datang ke B dari A menemukan jejak intensitas 15 di jalan yang mengarah ke H diletakkan oleh 15 semut yang berjalan seperti itu dari B, dan jejak intensitas 30 pada jalur C, diperoleh sebagai penjumlahan dari jejak diletakkan oleh 15 semut yang berjalan dari B dan oleh 15 semut yang mencapai B yang datang dari D melalui C. Probabilitas memilih jalan karena itu menjadi bias, sehingga jumlah semut yang diharapkan menuju C akan menjadi dua kali dari semut yang menuju ke H, yaitu 20 semut berbanding 10 semut. Hal yang sama berlaku untuk 30 semut baru di D yang datang dari E. Proses ini berlanjut sampai semua semut akhirnya akan memilih jalur terpendek



Gambar 2. Contoh dengan semut buatan

Penjelasan Gambar 2 dengan semut buatan adalah sebagai berikut:

- 1) Grafik awal dengan satuan jarak.
- 2) Pada waktu  $t = 0$  tidak ada jejak di tepi grafik, karena itu semut bebas memilih, apakah akan berbelok ke kanan atau kiri dengan probabilitas yang sama.
- 3) Pada waktu  $t = 1$  jejak lebih kuat pada lintasan-lintasan yang lebih pendek, karena lebih disukai oleh semut. Idenya adalah bahwa jika pada suatu titik semut harus memilih antara jalan yang berbeda, maka semut akan memilih jalur-jalur yang lebih dipilih oleh semut sebelumnya, karena zat feromon lebih dominan. Selanjutnya tingkat jejak tinggi adalah sama dengan jalur pendek [10].

Algoritma *Ant* dasar dapat dituliskan sebagai berikut [9]:

```
For each colony do
  For each ant do
    Generate route
    Evaluate route
    Evaporate pheromone in trails
    Deposit pheromone on trails
  End for
End for
```

Jadi cara kerja Algoritma *Ant* adalah sebagai berikut:

- 1) Pada awalnya, semut berkeliling secara acak.
- 2) Ketika semut-semut menemukan jalur yang berbeda, misalnya sampai pada persimpangan, mereka akan mulai menentukan arah jalan secara acak.
- 3) Sebagian semut memilih berjalan ke atas dan sebagian lagi akan memilih berjalan ke bawah.
- 4) Ketika menemukan makanan, maka mereka kembali ke koloninya sambil memberikan tanda dengan jejak feromon.
- 5) Karena jalur yang ditempuh lewat jalur bawah lebih pendek, maka semut yang bawah akan tiba lebih dulu dengan asumsi kecepatan semua semut adalah sama.
- 6) Feromon yang ditinggalkan oleh semut di jalur yang lebih pendek aromanya akan lebih kuat dibandingkan feromon di jalur yang lebih panjang.
- 7) Semut-semut lain akan lebih tertarik mengikuti jalur bawah karena aroma feromon lebih kuat [10].

Jenis-jenis algoritma *Ant*:

- 1) *Ant System*, merupakan algoritma ACO pertama yang dirumuskan dan diuji untuk menyelesaikan kasus TSP. Algoritma ini tersusun atas sejumlah semut yang bekerja sama dan berkomunikasi secara tidak langsung melalui komunikasi feromon. Cara kerja AS sebagai berikut: setiap semut memulai perjalanannya melalui sebuah titik yang dipilih secara acak (setiap semut memiliki titik awal yang berbeda). Secara berulang kali, satu persatu titik yang ada dikunjungi oleh semut dengan tujuan untuk menghasilkan sebuah jalur perjalanan [10].
- 2) *Elitist Ant System*, merupakan hasil pengembangan pertama dari AS, muncul berawal ketika adanya penguatan feromon pada lintasan-lintasan yang merupakan rute terbaik yang ditemukan sejak awal algoritma semut. Perjalanan terbaik ini dinotasikan dengan  $T_{bs}$  (*best-so-far tour*) [10] dan [11].
- 3) *Rank-Base Ant System* ( $AS_{Rank}$ ), merupakan pengembangan selanjutnya dari AS dan menerapkan *elitist strategy* [12]. Pada konsep peringkat dapat diterapkan dan dikembangkan pada AS sebagai berikut: setelah semua semut ( $m$ ) menghasilkan perjalanan, semut-semut diurutkan berdasarkan panjang perjalanan ( $L_1 \leq L_2 \leq \dots \leq$

Lm), dan kontribusi dari seekor semut dipertimbangkan menurut peringkat  $\mu$  (indeks peringkat) yang diberikan kepada semut tersebut. Sebagai tambahan bahwa hanya  $\omega$  terbaik yang dipertimbangkan [13] dan [14].

- 4) *MAX-MIN Ant System* (MMAS), merupakan pengembangan dari algoritma AS selanjutnya dengan dilakukan beberapa perubahan utama. Perubahan utamanya dengan memberikan batasan dalam pemberian nilai feromon dengan interval  $[\tau_{min}, \tau_{max}]$  [10].
- 5) *Ant Colony System*, merupakan pengembangan dari AS selanjutnya, setelah beberapa algoritma di atas. Algoritma ini tersusun atas sejumlah  $m$  semut yang bekerja sama dan berkomunikasi secara tidak langsung melalui komunikasi feromon.

Masalah penjadwalan perkuliahan dalam tulisan ini akan dianalisis dengan jenis *Rank-based Ant System*. Saat melakukan *update* feromon hanya  $(\omega-1)$  semut terbaik dan semut yang memiliki solusi *best-so-far* yang diperbolehkan meninggalkan feromon. Semut yang ke- $z$  terbaik memberikan kontribusi feromon sebesar  $\max\{0, \omega - z\}$ , sementara jalur perjalanan *best-so-far* memberikan kontribusi feromon paling besar, yaitu sebanyak  $\omega$ , dimana  $w$  adalah parameter yang menyatakan adanya perjalanan terbaik dan  $z$  adalah peringkat semut. Berikut ini adalah aturan transisinya:

$$P_{rs}^k = \begin{cases} \frac{[\tau_{rs}]^\alpha [\eta_{rs}]^\beta}{\sum [\tau_{ru}]^\alpha [\eta_{ru}]^\beta} & \text{untuk } s \in J_r^k \\ 0 & \text{untuk } s \text{ lainnya} \end{cases} \dots\dots\dots(1)$$

Dalam  $AS_{rank}$  aturan *update* feromonnya adalah sebagai berikut:

$$\tau_{rs} = (t + 1) = \rho \tau_{rs}(t) + \Delta \tau \dots\dots\dots(2)$$

Hasil dari evaluasi eksperimen oleh [13] menunjukkan  $AS_{rank}$  mempunyai hasil yang lebih baik daripada EAS dan lebih signifikan daripada AS. Di bawah ini dituliskan algoritma *Rank Based Ant System* [15]:

```

Initialize
For t=1 to number of iterations do
  For k = 1 to m do
    Repeat until ant k has completed a tour
      Select the time slot to be placed
      With probability  $p_{rs}$  given equation (1)
      Calculate the length  $L_k$  of the tour generated by ant k
    Update the trail levels  $\sigma_{rs}$  on all edges according to equation (2)
  End
```

Alberto Colorni, Marco Dorigo dan Vittorio Maniezzo melakukan penelitian algoritma genetika untuk menyelesaikan masalah penjadwalan. Masalah yang dipilih adalah penentuan kelas dengan *multi constraint*, *NP-hard*, dan masalah optimasi kombinasi untuk aplikasi dunia nyata. Yang pertama dilakukan adalah mendefinisikan struktur secara hirarki yang bertujuan untuk menentukan operator algoritma genetika yang sesuai dengan matriks yang menggambarkan jadwal. Jadwal yang akan dibuat adalah jadwal untuk sekolah. Algoritma genetika yang digunakan memiliki dua versi sebagai perbandingan, yaitu menggunakan atau tidak menggunakan *local search* [11].

Penelitian tentang masalah penjadwalan mata kuliah dilakukan oleh Socha, Sampels dan Manfrin menggunakan dua algoritma *Ant*, yaitu *Ant Colony* dan *Max-Min Ant System*. Tujuan dari penelitian ini adalah untuk menampilkan bagaimana algoritma *Ant* digunakan pada masalah dengan batasan yang banyak seperti penjadwalan

perkuliahan, dan menganalisis pengaruh dari pemilihan jenis tertentu dari algoritma *Ant* [14].

Selanjutnya Gang Wang, Wenrui Gong, dan Ryan Kastner dari UCLA juga meneliti penggunaan algoritma *Ant* untuk menyelesaikan masalah penjadwalan. Penjadwalan yang diteliti adalah masalah penjadwalan instruksi untuk pemetaan sebuah aplikasi ke sebuah peralatan komputasi. Karena masalah penjadwalan instruksi termasuk masalah *NP-hard*, maka harus menemukan metode yang efektif untuk menyediakan penyelesaian penjadwalan yang kualitatif. Oleh karena itu, digunakan algoritma MMAS yang sudah dikombinasikan dengan metode pencarian yang lain, yaitu *list scheduling*. Dengan kombinasi ini dapat dicapai hasil yang mendekati optimal dalam pencarian solusi [10].

Penelitian tentang penjadwalan mata kuliah juga dilakukan oleh Djamarus dan Mahamud dengan menggunakan algoritma *Ant* dengan feromon negatif, kemudian hasilnya dibandingkan dengan algoritma *Ant* yang tidak digabungkan dengan feromon negatif. Penelitian perbandingan dilakukan terhadap jadwal perkuliahan dan jadwal ujian [15].

Kemudian Lutuks dan Pongcharoen mengenalkan varian baru dari ACO yang disebut *Best-Worst Ant Colony System* (BWACS) untuk menyelesaikan masalah penjadwalan mata kuliah, mendemonstrasikan penggunaan desain dan analisis secara eksperimen untuk menyelidiki pengaturan parameter yang tepat dari BWACS ini [2].

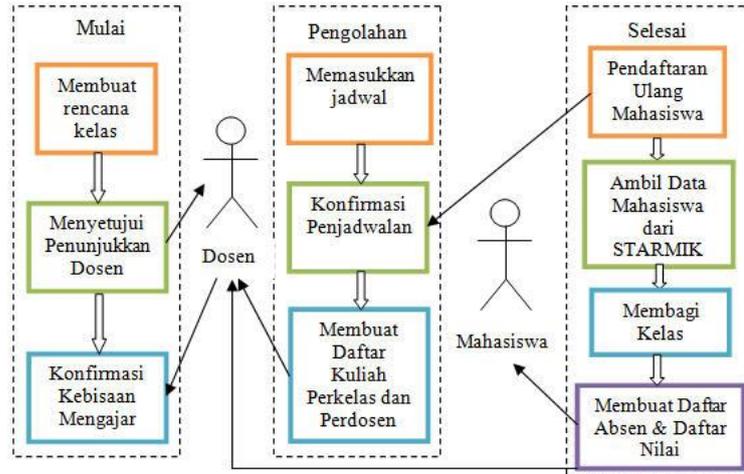
Berdasarkan pemaparan di atas, maka penulis melakukan analisis perangkat lunak penjadwalan mata kuliah menggunakan algoritma *ant* dengan jenis *Rank-based Ant System* untuk memecahkan masalah penjadwalan perkuliahan di STMIK Widya Dharma.

### 3. PERANCANGAN SISTEM

Prosedur proses penyusunan jadwal mata kuliah dan dosen yang berlaku saat ini adalah pembantu Ketua Bidang Akademik membuat rencana kelas berdasarkan data semester-semester sebelumnya. Rencana kelas ini berupa perkiraan jumlah kelas untuk mata kuliah tertentu sesuai dengan kurikulum yang berlaku. Melalui rapat pimpinan akan ditetapkan dan disahkan oleh masing-masing ketua program studi untuk menjadi laporan rencana perkuliahan yang berisi mata kuliah dan jumlah kelasnya, serta dosen pengasuh mata kuliah tersebut. Rapat ini biasanya dilaksanakan setelah UTS semester sebelumnya. Berdasarkan laporan yang sudah disahkan tersebut, maka pembantu ketua bidang akademik akan melakukan konfirmasi dengan dosen pengasuh mata kuliah mengenai waktu, baik hari maupun jam maupun mata kuliah yang bersedia untuk dijadwalkan. Setelah seluruh konfirmasi masuk, maka akan dibuatkan tabelnya dengan *Microsoft Excel*. Jika terdapat jadwal yang bertabrakan, maka akan dikonfirmasi ulang dengan dosen bersangkutan. Hal ini dilakukan sampai semuanya selesai dijadwalkan tanpa ada yang bertabrakan.

Proses selanjutnya adalah membuat daftar lengkap rencana perkuliahan per masing-masing dosen yang akan disampaikan kepada setiap dosen beserta dengan surat keputusan pengasuh mata kuliah yang ditandatangani oleh ketua STMIK Widya Dharma. Hal ini disampaikan sebelum rapat pertemuan semua dosen di lingkungan Widya Dharma, sehingga jika masih ada perubahan dapat disampaikan sebelumnya, agar dapat diumumkan sewaktu rapat pertemuan.

Penyusunan jadwal ini dapat berubah jika jumlah mahasiswa yang mengambil suatu mata kuliah tidak sesuai dengan jumlah yang diperkirakan, atau permintaan dari dosen untuk mengubah jam mengajar.



Gambar 3. Sistem penjadwalan berjalan

Penerapan algoritma *Ant* pada pengisian jadwal dosen pengasuh mata kuliah dapat diuraikan analoginya sebagai berikut: dosen adalah *ant*-nya, *slot* waktu dan ruang yang tersedia adalah koloninya, waktu yang tersedia, sedangkan mata kuliah yang diadakan dan kelas yang dibuka adalah jalurnya.

Sesuai dengan analisis untuk masalah penjadwalan, maka akan lebih sesuai jika menggunakan metode *rank based ant system*/RBAS. Dosen dengan peringkat yang lebih tinggi dapat dianalogikan dengan *Ant* yang mempunyai tingkat feromon yang lebih tinggi, sehingga mempunyai prioritas yang lebih tinggi. Pemberian peringkat berdasarkan penguasaan materi mata kuliah dan waktu kebiasaan masing-masing dosen. Maksud dari penguasaan materi mata kuliah adalah banyaknya dosen yang dapat mengasuh suatu mata kuliah, semakin sedikit dosen yang menguasai mata kuliah semakin tinggi prioritasnya yang berkorelasi dengan peringkat yang lebih tinggi. Untuk faktor waktu kebiasaan dosen, pemberian peringkat disesuaikan dengan tingkat ketersediaan waktu masing-masing dosen, artinya dosen dengan waktu kebiasaan sedikit, maka mempunyai peringkat yang lebih tinggi. Setelah pemberian peringkat selesai dilakukan untuk semua dosen, maka penjadwalan baru dapat dimulai. Dosen dapat memilih *slot* waktu secara bebas, dengan catatan setiap *slot* waktu hanya boleh dipilih sekali. *Slot* waktu yang mungkin terpilih adalah yang lebih disukai pada umumnya dan biasanya seorang dosen akan memilih *slot* waktu yang berdekatan agar efisien dalam melakukan perjalanan ke kampus.

Berdasarkan algoritma RBAS, maka komponen-komponen dari RBAS untuk menyelesaikan masalah penjadwalan dapat digambarkan berikut:

- 1) *Ant* adalah dosen yang disusun berdasarkan peringkat (*ranking*).
- 2) *Colony* adalah *slot* waktu dan ruang kuliahnya.
- 3) *Route* adalah ruang kuliah sesuai dengan mata kuliah yang cocok dengan dosen yang memiliki waktu sesuai dengan *colony*.
- 4) *Elite* adalah dosen yang memiliki waktu yang cocok dalam *colony* yang berupa *local optimum*.
- 5) *Elite trail* adalah jalur yang di-plot sesudah adanya *elite*.
- 6) *Best global route* adalah semua jalur terbaik yang berhasil di-plot.

Maka keterangan untuk persamaan (1) dalam analisis penjadwalan adalah:

- $k$  adalah individu dosen.
- $\tau_{rs}$  adalah intensitas feromon dari lintasan antara *slot* waktu.
- $\alpha$  adalah sebuah parameter yang mengontrol bobot (*weight*) relatif dari feromon.

- $\eta_{rs} = \frac{1}{d_{rs}}$  adalah visibilitas dari *slot* waktu dimana  
 $d_{rs} = \sqrt{(x_r - x_s)^2 + (y_r - y_s)^2}$  (jika hanya diketahui koordinat titiknya saja).
- $\beta$  adalah parameter pengendali jarak ( $\alpha > 0$  dan  $\beta > 0$ ).
- $\Omega = I_r^k$  adalah jumlah *slot* waktu yang belum terpilih.

Dengan cara ini kita memilih lintasan yang lebih pendek dan memiliki jumlah feromon yang lebih besar. Jadi proses ini dilakukan berulang untuk tiap k (dosen) sampai semua dosen menyelesaikan penugasannya. Pada tiap langkah dari iterasi t kumpulan dari *slot* waktu akan dikurangi satu persatu hingga akhirnya tinggal satu *slot* waktu yang tersisa dengan probabilitas  $p_{rs}$  adalah satu. Kemudian jejak di-update sesuai dengan tingkat prioritas yang diberikan menurut analoginya, maka setiap *slot* waktu terisi berlaku  $(1-p)$  sebelum *slot* waktu berikutnya dikerjakan.

Keterangan untuk *update trail*-nya pada persamaan (2) adalah:

- t = iterasi penghitung
- $\rho \in [0,1]$  parameter yang mengatur pengurangan dari  $\tau_{rs}$
- $\Delta \tau_{rs}$  jumlah penambahan jejak
- m = jumlah dosen
- $\Delta \tau_{rs}^{\omega k}$  = penambahan jejak oleh dosen ke-k
- Q kuantitas dari feromon dosen/*slot* waktu
- $L^k$  banyaknya *slot* waktu dosen

Analisis dimulai dari pemasukan data yang dibutuhkan oleh sistem sebagai bahan baku dan setelah diproses dikeluarkan sebagai informasi. Sebagai masukan (*input*) adalah:

- 1) Kelas yang diberikan kepada masing-masing dosen untuk mata kuliah tertentu.
- 2) Jadwal yang dapat diisi oleh seorang dosen pada hari dan jam tertentu.
- 3) Jadwal yang tidak boleh diisi.
- 4) Ruang yang dapat dipergunakan untuk perkuliahan.
- 5) Jumlah mahasiswa yang melakukan pendaftaran ulang.

Informasi (*output*) yang dihasilkan adalah:

- 1) Daftar dosen.
- 2) Daftar mata kuliah.
- 3) Daftar pengajaran per dosen.
- 4) Daftar pengajaran per mata kuliah.
- 5) Daftar dosen penanggung jawab mata kuliah.
- 6) Daftar waktu kebiasaan dosen per dosen.
- 7) Daftar waktu kebiasaan dosen per sesi.
- 8) Laporan jadwal kelas.
- 9) Laporan jadwal laboratorium.
- 10) Laporan jadwal dosen.
- 11) Daftar informasi transfer jadwal.

Perspektif dari perangkat lunak ini adalah merupakan *software* penjadwalan otomatis berdasarkan mata kuliah, dosen, ruang, jadwal, dan mahasiswa. Perangkat lunak ini menyediakan pengisian data melalui papan ketik, kemudian dengan menekan tombol tertentu, maka jadwal akan terisi secara otomatis. Selain mengisi *slot* secara otomatis, juga dapat melakukan optimasi dalam penjadwalan dengan mengurangi kesalahan pada:

- Adanya pengisian *slot* waktu oleh dua atau lebih perkuliahan yang berbeda.
- Dosen yang mengajar pada waktu bersamaan.
- Perkuliahan yang diisi *slot*-nya tidak sesuai dengan jadwal dosen.
- Jumlah siswa melebihi kapasitas ruang.
- Jumlah jadwal dosen tidak sesuai dengan jumlah kelas yang diberikan.

- Mahasiswa mengikuti lebih dari satu mata kuliah pada waktu yang sama.  
Fungsi utama dari perangkat lunak ini adalah:
- Mengelola semua data yang berhubungan dengan penjadwalan, yaitu mata kuliah, dosen, ruang, jadwal kuliah, jadwal dosen, dan kelas dosen.
- Mengisi *slot* jadwal secara otomatis dan menambahkan jadwal yang sudah ada.
- Membuat laporan jadwal per hari, per dosen, dan per mata kuliah.
- Mengatur perubahan jadwal.

Perangkat lunak yang dirancang mempunyai beberapa batasan, seperti pengambilan data hanya yang menyangkut penjadwalan saja, seperti dosen, mata kuliah dan ruang, serta waktu perkuliahan. Masalah keuangan dan nilai ujian tidak dibahas karena belum dimasukkan ke dalam sistem ini.

Asumsi adalah suatu pemisalan, jadi berupa kesalahan yang dianggap benar dengan suatu alasan. Beberapa asumsi adalah:

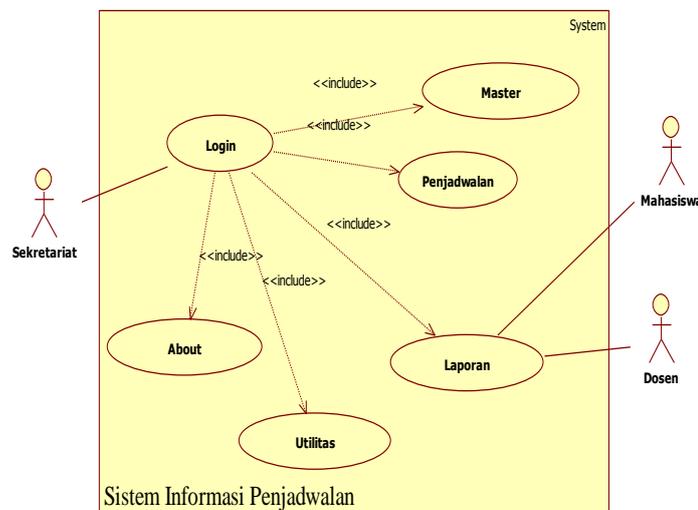
- 1) Data yang dimasukkan adalah data yang bersih (bebas *noise*, redundansi, dan *mission value*).
- 2) Semua mahasiswa tidak memiliki batasan waktu, maksudnya semua waktu mahasiswa adalah kosong, sehingga dapat diisi dengan mata kuliah tertentu.
- 3) Pembagian waktu berdasarkan pada durasi 2 SKS ataupun 1,5 SKS.

Tabel 1. Definisi aktor

No.	Aktor	Deskripsi
1.	Sekretariat	Pemakai aplikasi yang bertugas mengatur jadwal serta mengumpulkan semua data yang berkaitan dengan jadwal.
2.	Dosen	Orang yang mengasuh mata kuliah pada mahasiswa, membutuhkan laporan jadwal mengajar.
3.	Mahasiswa	Orang yang mengambil mata kuliah, membutuhkan laporan jadwal kuliah.

Diagram *use-case* berikutnya menggambarkan kebutuhan sistem yang meliputi:

- 1) Penentuan aktor (entitas yang terlibat dengan sistem).
- 2) Penentuan *use case* (aktivitas yang dikerjakan oleh sistem).
- 3) Skenario (cara kerja dari setiap *use case*).
- 4) Diagram *use case* (menggambarkan hubungan *use case* dan aktor dalam diagram).

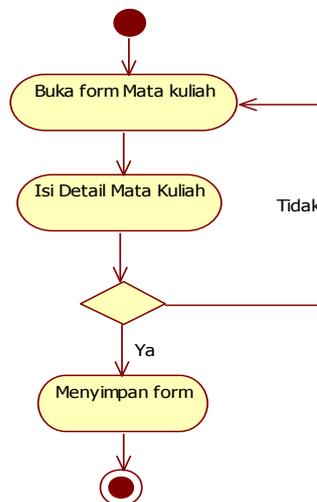


Gambar 4. Diagram *use case* sistem informasi penjadwalan

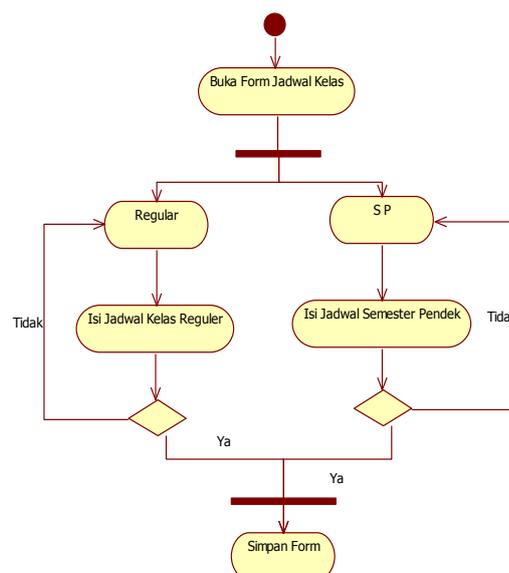
Perlu dijelaskan bahwa dalam tiap *use case* masih terdapat sub menunya, yaitu:

- 1) Menu *Master*, terdiri dari Mata Kuliah, Dosen, Jurusan, Kelas, Waktu Dosen, Waktu Laboratorium, dan Keluar
- 2) Menu Penjadwalan, terdiri dari Dosen-Matkul, Dosen-Tg. Jawab, Jadwal Kelas (Regular dan SP), Jadwal Lab., transfer Jadwal dan Cek Waktu Kritis
- 3) Laporan, terdiri dari Daftar Dosen, Daftar Mata Kuliah, Daftar Pengajaran (per dosen dan per mata kuliah), Daftar Dosen Penanggung Jawab, Daftar Waktu Dosen (per dosen dan per sesi waktu), Daftar Jadwal Kelas, Daftar Jadwal Lab., Daftar Jadwal Dosen, dan Daftar Informasi Transfer Jadwal.
- 4) Utilitas, terdiri dari: *Re-Index*, *Pack Data*, *Backup Jadwal*, Pengaturan Awal Tabel, Ubah *Password*, dan Ubah Waktu Kuliah

Kemudian akan digambarkan *activity diagram* yang menggambarkan berbagai aliran kegiatan yang ada dalam perancangan sistem, mulai dari aliran bermula, keputusan yang mungkin terjadi, dan bagaimana aliran berakhir. Setiap *use case* akan digambarkan *activity diagram*-nya. Di bawah ini digambarkan dua diantaranya, yaitu *activity diagram* Mata Kuliah dan *activity diagram* Jadwal Kelas.

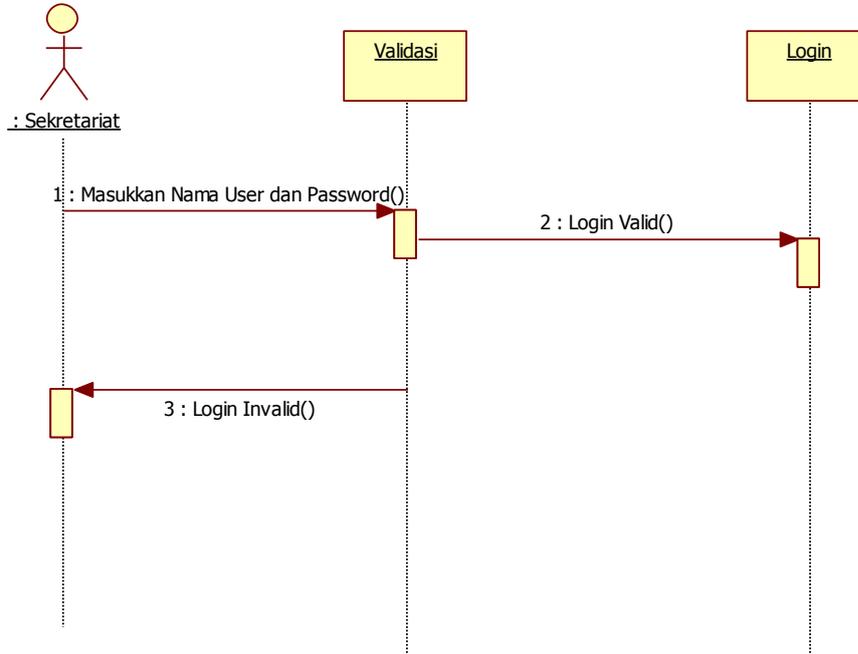


Gambar 5. *Activity diagram* mata kuliah

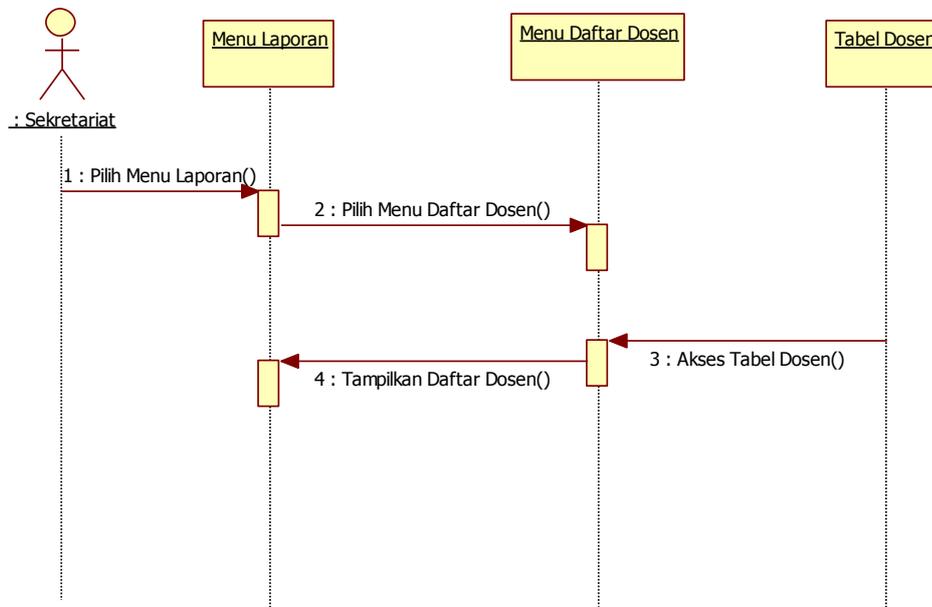


Gambar 6. *Activity diagram* jadwal kelas

Selanjutnya akan digambarkan *sequence diagram* yang merupakan gambaran interaksi antar objek yang disusun menurut urutan waktu, yaitu urutan kejadian yang dilakukan oleh seorang aktor dalam menjalankan sistem. Berikut ini disajikan dua *sequence diagram*, yaitu *sequence diagram login* dan *sequence diagram* daftar dosen.

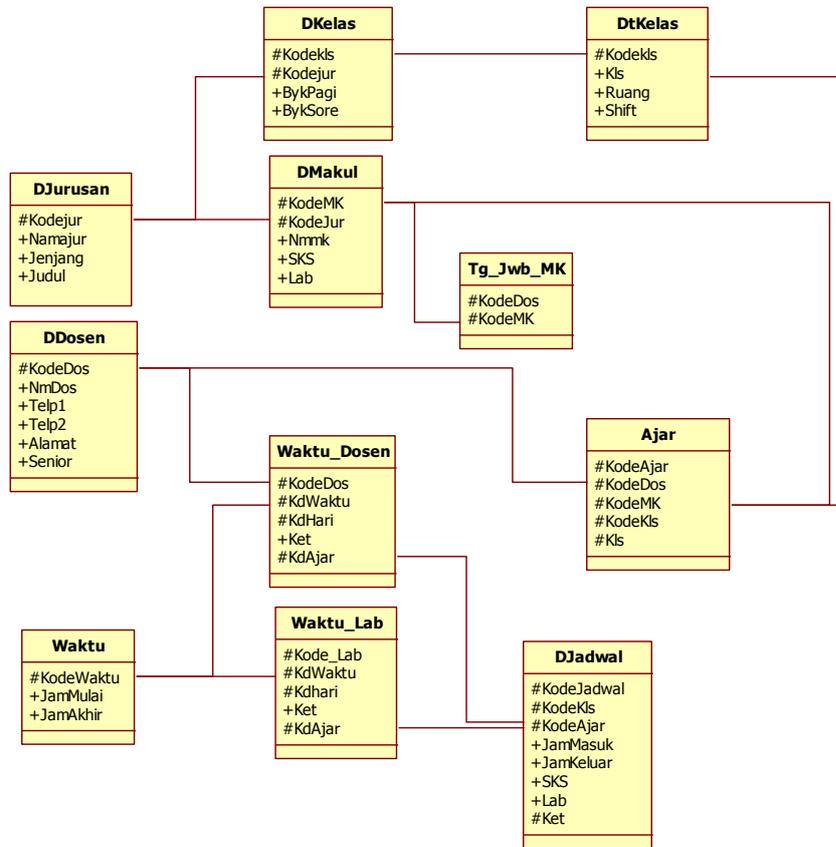


Gambar 7. *Sequence diagram login*



Gambar 8. *Sequence diagram* daftar dosen

Rangkaian UML terakhir yang digambarkan dalam tulisan ini adalah *Class Diagram* yang digambarkan sebagai berikut:

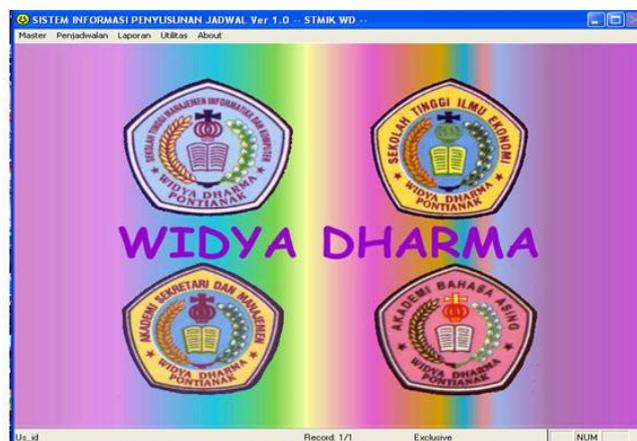


Gambar 9. *Class diagram* sistem informasi penjadwalan

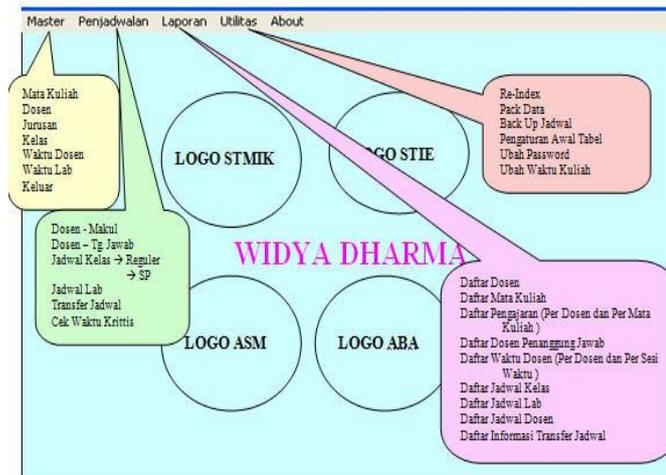
#### 4. HASIL DAN PEMBAHASAN

##### 4.1 Rancangan dan Tampilan dari Sistem Aplikasi

Sesuai dengan perancangan di atas, maka berikut ini akan ditampilkan beberapa dari *form* rancangan dan tampilan sewaktu dijalankan.



Gambar 10. Rancangan menu utama



Gambar 11. Tampilan menu utama



Gambar 12. Tampilan menu mata kuliah



Gambar 13. Tampilan menu waktu dosen

Gambar 14. Tampilan menu jadwal kelas

Gambar 15. Tampilan menu penyusunan jadwal kelas

## 4.2 Pembahasan dari Hasil Rancangan

Gambar 14 merupakan pusat dari pengolahan data yang telah dimasukkan dari awal. Bentuk *form* ini dirancang sesuai dengan format yang berlaku sekarang, dengan dasar pemikiran untuk memudahkan pemakai yang ada di sekretariat yang sudah terbiasa dengan bentuk yang ada. *Form* dimulai dengan penentuan jurusan, semester, dan kelas yang akan disusun jadwalnya. Peng-*input*-an jadwal dilakukan dengan menekan tombol bertuliskan ‘JAM I’ atau ‘JAM II’ atau ‘JAM III’ yang tersedia dimana telah dikelompokkan berdasarkan hari. Penekanan tombol ini memunculkan layar kecil seperti gambar di kanan bawah (Gambar 15) sebagai tempat pengisian detail daripada jadwal. Dosen-dosen yang telah ditunjuk mengajar akan dipilih di sini. Dosen dan mata kuliah yang muncul telah diverifikasi melalui waktu dosennya sehingga memudahkan dalam pemilihan.

Fasilitas lainnya adalah pengecekan dosen dengan waktu kritis dan pengecekan dosen dengan mata kuliah mana yang belum terjadwal. Pengecekan waktu kritis dosen akan memberikan informasi tentang dosen dan mata kuliah mana yang harus diprioritaskan mengingat keterbatasan waktu kebiasaan (tingkat feromon tinggi). Pengecekan mata kuliah yang belum masuk jadwal akan membantu agar tidak ada jadwal yang tertinggal untuk disusun.

Setiap pemasukan jadwal akan langsung mempengaruhi tabel waktu dosen yang telah ada menjadi terisi sehingga pada waktu tersebut dosen tidak dapat melakukan pengisian jadwal di kelas yang berbeda.

## 5. KESIMPULAN

Masalah penjadwalan dapat didekati dengan metode meta heuristik, dalam tulisan ini digunakan algoritma *Ant* yang merupakan salah satu metode meta heuristik yang cukup banyak dipergunakan. Algoritma *Ant* jenis  $AS_{Rank}$  dapat menghasilkan suatu jadwal yang baik. Kendala utama dari penyusunan jadwal perkuliahan di STMIK Widya Dharma adalah perbandingan yang tidak seimbang antara jumlah kelas yang banyak dengan jumlah dosen yang terbatas, sehingga memperbesar kemungkinan terjadinya bentrok jadwal. Aplikasi yang dirancang dan dikembangkan ini masih perlu untuk dikembangkan lebih lanjut, misalnya dapat secara langsung menangkap data mahasiswa yang mendaftar pada STARMIK, sehingga dapat melakukan penyusunan absensi dan daftar nilai secara otomatis juga.

## REFERENSI

- [1]. Jain Ashish, Jain Dr. Suresh, and Chande Dr. P.K., “*Formulation of Genetic Algorithm to Generate Good Quality Course Timetable*”, International Journal of Innovation, Management and Technology, Vol. 1, No. 3, pp. 248-251, August 2010.
- [2]. Lutuksin Thatchai and Pongcharoen Pupong, “*Best-Worst Ant Colony System Parameter Investigation by Using Experimental Design and Analysis for Course Timetabling Problem*”, Second International Conference on Computer and Network Technology, IEEE Computer Society, pp. 467-471, 2010.
- [3]. Karl F.Doerner, Daniel Merkle, and Thomas Stützle, “*Special Issue on Ant Colony Optimization*”, Swarm Intell (2009) 3: 1-2, DOI 10.1007/s11721-008-0025-1.
- [4]. Pei Hua Chen and Hua Hua Cheng, “*IRT-based Automated Test Assembly: A Sampling and Stratification Perspective*”, The University of Texas at Austin, August 2005.
- [5]. Cole A. J., “*The Preparation of Examination Time-tables Using A Small-Store Computer*”, Computer Journal, 7: 117-121, 1964.
- [6]. Welsh D.J.A. and Powell M. B., “*An Upper Bound for The Chromatic Number of A Graph and Its Application to Timetabling Problems*”, Computer Journal, 10(1): PP. 85-86, 1967.
- [7]. Sadaf N. Jat and Yang Shengxiang, “*A Memetic Algorithm for the University Course Timetabling Problem*”, 20<sup>th</sup> IEEE International Conference on Tools with Artificial Intelligence, DOI 10.1109/ICTAI.2008.126, pp. 427-433, 2008.
- [8]. Hsio- Lang Fang, “*Genetic Algorithms in Timetabling and Scheduling*”, Department of Artificial Intelligence University of Edinburgh, 1994.
- [9]. Solnon Christine, “*Ants Can Solve Constraint Satisfaction Problems*”, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 4, pp. 347-357, August 2002.
- [10]. Marco Dorigo and Alberto Coloni, “*The Ant System: Optimization by A Colony of Cooperating Agents*”, IEEE Transaction on Systems, Man, and Cybernetics-Part B, Vol. 26, No. 1, pp. 1-13,1996.
- [11]. Alberto Coloni, Marco Dorigo, Vittorio Maniezzo, “*Genetic Algorithms and Highly Constrained Problems: The Time-Table Case*”, Proceedings of the First International Workshop on Parallel Problem Solving from Nature, Springer-Verlag, pp. 55-59.
- [10]. Wang Gang, Gong Wenrui, and Kastner Ryan, “*Instruction Scheduling Using Max-Min Ant System Optimization*”, GLSVLSI’05, Chicago, Illinois, USA, April 17-19, 2005.

- [11]. Alberto Colorni, Marco Dorigo, Vittorio Maniezzo, “*A Genetic Algorithm To Solve The Timetable Problem*”, Submitted To Computational Optimization And Applications Journal, 1993.
- [12]. Bernd Bullnheimer, Richard F. Hartl, and Christine Strauß, “*A New Rank Based Version of the Ant System-A Computational Study*”, Working Paper Series No. 1, April 1997.
- [13]. Bullnheimer, Hartl, and Strauss, “*An Ant Colony Optimization Approach for the Single Machine Total Tardiness Problem*”, Department of Management Science University of Vienna, 1999.
- [14]. Socha Krzysztof, Sampels Michael, and Manfrin Max, “*Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art*”, IRIDIA, Université Libre de Bruxelles, CP 194/6.
- [15]. Djasli Djamarus and Ku Ruhana Ku-Mahamud, “*Ant System Algorithm with Negative Pheromone for Course Scheduling Problem*”, Eighth International Conference on Intelligent Systems Design and Applications, DOI 10.1109/ISDA.2008.154, 2008.